AD-A091 521   CARNEGIE-MELLON UNIV PITTSBURGH PA MANAGEMENT SCIENC--ETC F/G 12/1
THE SOLUTION OF MANPOWER PLANNING PROBLEMS BY THE FORWARD SIMPL--ETC(U)
AUG 80 J E ARONSON, G L THOMPSON   N00014-75-C-0621
UNCLASSIFIED MSRR-463   NL

END
DATE FILMED
12 80
DTIC

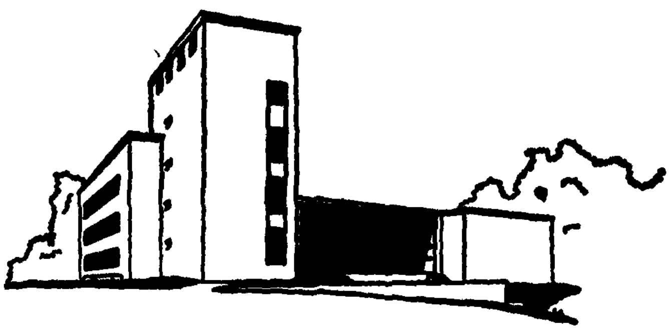
LEVEL II

Carnegie-Mellon University

PITTSBURGH, PENNSYLVANIA 15213

GRADUATE SCHOOL OF INDUSTRIAL ADMINISTRATION

WILLIAM LARIMER MELLON, FOUNDER



W.P. No. 3-80-81

Management Science Research Report No. 463

THE SOLUTION OF MANPOWER PLANNING PROBLEMS BY THE FORWARD SIMPLEX METHOD

by

Jay E. Aronson*

and

Gerald L. Thompson**

August, 1980

* Dept. of O.R. and Engineering Management
Southern Methodist University

** Graduate School of Industrial Administration
Carnegie-Mellon University

This research was sponsored by the Navy Personnel Research and Development Center (WR-6-0147) and the Office of Naval Research through Contract N00014-75-C-0621 with Carnegie-Mellon University. Reproduction in whole or in part is permitted for any purpose of the U. S. Government.

Management Sciences Research Group
Graduate School of Industrial Administration
Carnegie-Mellon University
Pittsburgh, Pennsylvania 15213

THE SOLUTION OF MANPOWER PLANNING PROBLEMS

BY THE FORWARD SIMPLEX METHOD

by

Jay E. Aronson and Gerald L. Thompson

ABSTRACT

The use of the forward simplex algorithm of Aronson, Morton, and Thompson to solve the multi-stage personnel planning linear programming models of Charnes, Cooper, and Niehaus is described. Computational Results on randomly generated problems having up to 200 periods indicate that the forward simplex method requires CPU time and number of pivots which are linear in the number of periods. The standard simplex method requirements vary with at least the cube of the number of periods. For this reason the forward simplex method should be especially useful for solving real-time, conversational versions of personnel (and other) planning models.

KEY WORDS

Forward simplex method

Staircase structured linear programs

Large scale linear programs

1. Introduction

The Forward Simplex Method due to Aronson, Morton, and Thompson [1,3] is an adaptation of the ordinary simplex method for solving general dynamic (staircase) linear programs. Such models typically occur in problems in which it is necessary to plan over time, and commonly occur in the management of personnel, production, energy, and economic systems.

In the present paper we discuss the application of the forward simplex method to the solution of the manpower planning models originally developed by Charnes, Cooper, and Niehaus [5,6]. It would also be applicable (although we have not specifically tested it) to solving in real time the conversational version of these models discussed by Niehaus, Sholtz, and Thompson [10,11].

A brief discussion of the forward simplex algorithm is given in Section 2 and a description of its computer implementation appears in Section 3. A summary of the manpower planning model and the computation tests made with it appear in Section 4.

A series of randomly generated problems having from 5 to 200 periods were solved. The largest (200 period) problem was solved in a little more than 2 seconds on a DEC-20 computer without matrix reinversions. Its solution by an ordinary simplex code would have involved a tableau of size 2600 rows and 2600 columns, which would have been a challenging problem for a standard LP code to solve. A 20 period problem required more than 1000 seconds for our standard LP code to solve. Regression results indicate that the number of pivots and CPU time required by the forward simplex code vary linearly with the number of periods.

2. The Forward Simplex Method

In this section we present a brief description of the Forward Simplex Method. For a more detailed discussion, the reader is referred to [3]. Consider the general staircase linear program:

$$
(1)\quad \begin{cases}
\text{(a)} \quad \min \sum_{t=1}^{T} c_t X_t \\
\qquad \text{subject to} \\
\text{(b)} \quad A_1 X_1 = d_1 \\
\text{(c)} \quad B_{t-1} X_{t-1} + A_t X_t = d_t \quad , \quad t = 2,\ldots,T \\
\text{(d)} \quad X_t \geq 0 \quad , \quad t = 1,\ldots,T
\end{cases}
$$

where c_t is 1 by n_t, A_t is m_t by n_t, B_t is m_{t+1} by n_t, d_t is m_t by 1, and X_t is n_t by 1. We assume for simplicity, $A_t = A$ and $B_t = B$ for $t = 1,\ldots,T$, where A and B are fixed matrices. This assumption can be easily relaxed. The matrices A and B are partitioned as described in [3].

Let (1) be a T-period subproblem of some longer problem with length T_p. The Forward Simplex Method first partially solves the 1-period subproblem. It then augments this solution to form an initial basic feasible solution to the 2-period subproblem, partially solves this one, and so on for T = 3 to T_p. (A partial solution restricts pass-on variables of the current subproblem period to be nonbasic at zero).

Augmentation and pivoting continue until either the T_p-period problem is solved, or the entire allowable tableau space fills up. When the tableau is full, the available tableau space is considered to be a small window into the entire problem. This tableau window is slid down and to the

right, discarding early stable data, and the newest data are augmented into the window. A wrap-around tableau feature is used instead of actually sliding the window. This is discussed in detail in [3]. This technique works because the Forward Simplex Method maintains as much as possible of the staircase structure of (1) in solving the problem. In fact, problems exceeding ten times the tableau window size were solved. The augmentation in a period, T, is performed by including the (T-1) period B matrix, and the T-period A matrix in the tableau. The entering column is chosen from right to left, the minimum ratio test is performed from bottom to top. The Forward Simplex Method is efficient because it exploits a natural decomposition [3] of the problem. No reordering of the rows and columns is necessary to maintain the staircase structure. Thus, there is an automatic spike reduction [8], [9]. Intuitively later period forecasts should not have much impact on early decisions. The natural decomposition of the staircase problem tends to isolate early decisions from the effects of later period decisions.

3. The Code

This first version of the Forward Simplex Method (FORLP) was written in FORTRAN and developed on the Carnegie-Mellon University DEC 20/60 B. It requires 256K of addressable core. The global data require 224K, leaving 32K for local data. The code can handle up to 5000 time periods. The maximum dimensions of the A and B matrices are 22 by 22. The tableau window is dimensioned 336 by 322, so that 14 periods of the largest A matrix fit. In the interest of programming in "standard" FORTRAN, all do loops increment. To achieve portability, only the statements that open and close disk files need be changed.

The code utilizes the condensed Tucker tableau for ease of implementation. This is a standard simplex tableau but without the identity matrix that corresponds to the basic columns. Later versions of the code will utilize a compact form of the inverse. As outlined in [3], an auxiliary variable (X^t_{n+1}) and constraint ($X^t_{n+1} \le 1$) are added to each period. In the case of equality constraints, one extra row is added, per period, to convert them to inequality constraints. A perturbed right hand side is used to handle degeneracy.

All input is read from a disk file. Output, at the user's option, is printed onto a disk file, or onto his terminal. When the tableau window is full the code searches for a heuristic planning horizon [3]. If one is found, the primal and dual solutions up to the planning horizon are printed onto two disk files, the appropriate tableau space is cleared, and augmentation continues. When no such horizon exists, the problem requires a larger window.

Data must be appropriately scaled, all lower bounds on variables must be zero, and the constraint matrices must be partitioned as described in [3].

No basis reinversion is employed in this version of the code. The natural decomposition of staircase models effectively blocks numerical errors from rippling from early periods into later ones, so that numerical difficulties were not encountered on the test problems tried.

For comparison purposes, a standard linear programming code (STDLP) was developed from FORLP. This code also utilizes a condensed Tucker tableau, but with a single auxiliary row and column, and if necessary, a single extra row for handling equality constraints. The first positive reduced cost rule is used to determine the entering variable. The standard LP code was used on all three models as a benchmark for the Forward Simplex Method. Results for both codes are reported. Unfortunately, due to space limitations the standard LP code could only solve problems with a maximum tableau size of 334 by 321. In spite of this space problem, STDLP was an adequate measure of the kind of performance expected from a standard LP code. Next the computational results are presented.

4. The Manpower Planning Model

Here, the performance of the Forward Simplex Method on a version of the manpower planning model in [10] is discussed. This is a simplified version of a goal programming model that deals with intake or recruiting requirements planning for the Naval Underwater System Center (NUSC), a large naval laboratory. There are two manpower grades, with specified transition probabilities of manpower from grade to grade and out of the system. The model is now presented.

The goals or requirements for grade i in period t are D_t^i. The on-board manpower of grade i in period t is E_t^i, the manpower over the goal (under time) is U_t^i, the manpower under the goal (overtime) is V_t^i, the number hired at the beginning of period t is H_t^i and the number fired is F_t^i. Lower and upper bounds on E_t^i are $\underline{E}_t^i$ and $\bar{E}_t^i$ respectively. The upper bound in-grade constraint limit in period t is $\bar{G}_t$, the upper bound on manpower requirements is $\bar{M}_t$, and the budget is B_t. The transition probabilities are given by p_t^{ij} for an employee's grade classification changing from grade i in period t-1 to grade j in period t. Penalties of h, f, u, and v are incurred per unit of H_t^i, F_t^i, U_t^i, and V_t^i required to meet the goal. The T period problem is stated as problem (M_T) in Figure 1.

$$
(M_T)\begin{cases}
\text{(a)} & \min \sum_{t=1}^{T} \sum_{i=1}^{2} (hH_t^i + fF_t^i + uU_t^i + vV_t^i) \\
& \text{subject to} \\
\text{(b)} & E_t^i - U_t^i + V_t^i = D_t^i \quad ; \quad i=1,2;t=1,\ldots,T \\
\text{(c)} & p^{1i}E_{t-1}^1 - p^{2i}E_{t-1}^2 + E_t^i - H_t^i + F_t^i = 0 \quad ; \quad i=1,2;t=1,\ldots,T \\
\text{(d)} & b_t^1E_t^1 + b_t^2E_t^2 \leq B_t \\
\text{(e)} & c_t^1E_t^1 + c_t^2E_t^2 \leq \bar{G}_t \\
\text{(f)} & E_t^1 + E_t^2 \leq \bar{M}_t \\
\text{(g)} & E_t^i \leq \bar{E}_t^i \quad i=1,2;t=1,\ldots,T \\
\text{(h)} & E_t^i \geq \underline{E}_t^i \quad ; \quad i=1,2;t=1,\ldots,T \\
\text{(i)} & E_t^i, U_t^i, V_t^i, H_t^i, F_t^i \geq 0
\end{cases}
$$

Figure 1. In this model, two of the ten variables are pass-ons.

For the model tested the costs and parameters were: $h = 1$, $f = 3$, $u = v = 2$, $p_t^{11} = .8$, $p_t^{12} = .1$, $p_t^{21} = .2$, $p_t^{22} = .7$, $b_t^1 = 8000$, $b_t^2 = 10{,}000$, $c_t^1 = 1$, and $c_t^2 = 2$. To make each T period subproblem feasible with the pass-on variables E_t^i nonbasic at zero, extra slack variables at high costs were added to constraint (M_Th). After adapting the matrices to the proper format, they were dimensioned 13 by 13. Only 24 periods fit into the tableau window.

Eleven problems with 20 time periods were generated. All right hand sides, except for ($M_T g$-h) were generated with a cyclic pattern of period six, plus a random component drawn from a uniform distribution. For each generated right hand side parameter, the cyclic patterns were offset by a fixed number of periods.

Both FORLP and STDLP were used to solve the eleven problems. The problem length T started at 5 and has incremented by 5, until the 20 period problems were solved. For problems solved with FORLP, the mean pivoting CPU time versus T is plotted in Figure 2. In Figure 3 the same results appear for problems solved with STDLP. The mean number of pivots versus T for problems solved with both FORLP and STDLP is plotted in Figure 4.

For the model, the regression analysis results are given in (2) and (3).

$$(2)\begin{cases} \text{CPU Time}_{\text{FORLP}} = -\ 332.818 + 124.673T & ; \quad R = .997 \\ \text{No. of Pivots}_{\text{FORLP}} = -\ 19.227 + 16.242T & ; \quad R = .999 \end{cases}$$

$$(3)\begin{cases} \text{CPU Time}_{\text{STDLP}} = 22.725T^{3.594} & ; \quad R = 1.000 \\ \text{No. of Pivots}_{\text{STDLP}} = -\ 365.306 + 112.453T & ; \quad R = .995 \end{cases}$$

At T = 20 the mean total pivoting time of FORLP was about .2% of that of STDLP (about 2 seconds compared to 1100). FORLP is linear in T, while STDLP is at least cubic. For this data set, STDLP was unbounded for one problem at T = 10, two at T = 20, three at T = 30 and 40, with one infeasible at T = 40.

For this problem set, FORLP required about 80% more time to perform

all extra overhead, disk I/O, and solution reconstruction at T = 20. STDLP required about .2% more time. Even with the overhead, FORLP is linear in T, and much more efficient than STDLP.

Next, eleven periods of 200 periods were generated with the same demand pattern. For this problem set solved with FORLP, the problem length was incremented by 25. Normally, a tableau dimensioned 2600 by 2600 would be required for a 200 period problem. Here the tableau window had dimensions of 312 by 312 (for 24 periods).

The mean pivoting CPU time versus T is plotted in Figure 5. In Figure 6, the mean number of pivots versus T is plotted. Since the wrap-around feature is used at T = 25, the first few points of the mean CPU time curve are included in this analysis. The regression results are stated in (4).

$$(4)\quad \begin{cases} \text{CPU Time}_{\text{FORLP}} = -3139.99 + 232.68T & ;\ R = 1.000 \\ \text{No. of Pivots}_{\text{FORLP}} = -15.383 + 15.373T & ;\ R = 1.000 \end{cases}$$

Both regressions are linear, the total solution CPU time was about 23% more than the pivoting CPU time at T = 200. FORLP solved the 200 period problems in about 43 seconds, much less than the 1093 seconds required by STDLP to solve the 20 period problems. Again, the performance of FORLP proved linear for both solution time and the number of pivots.

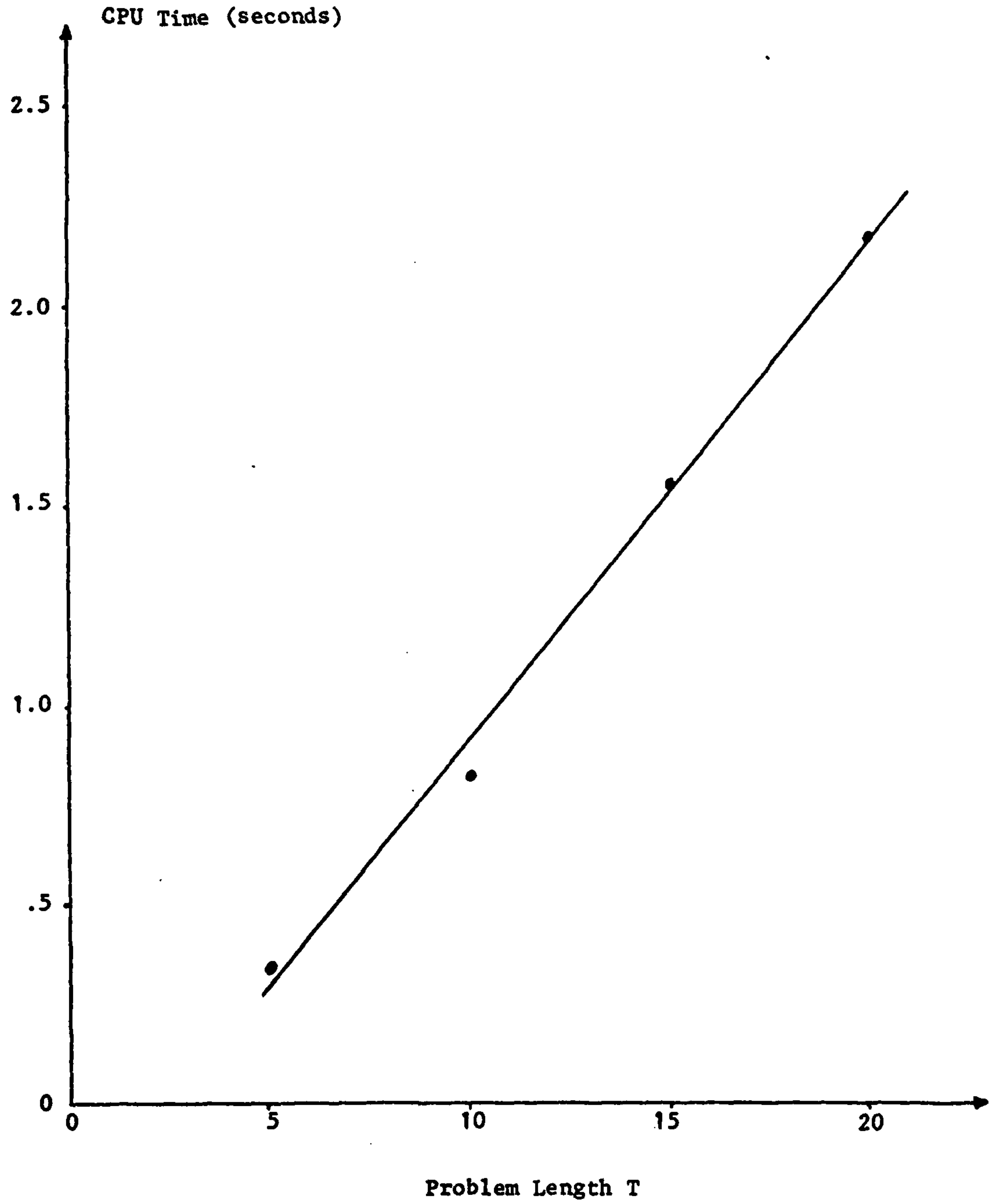


Figure 2: Mean Pivoting CPU Time versus T for 11 randomly generated manpower planning problems solved with FORLP

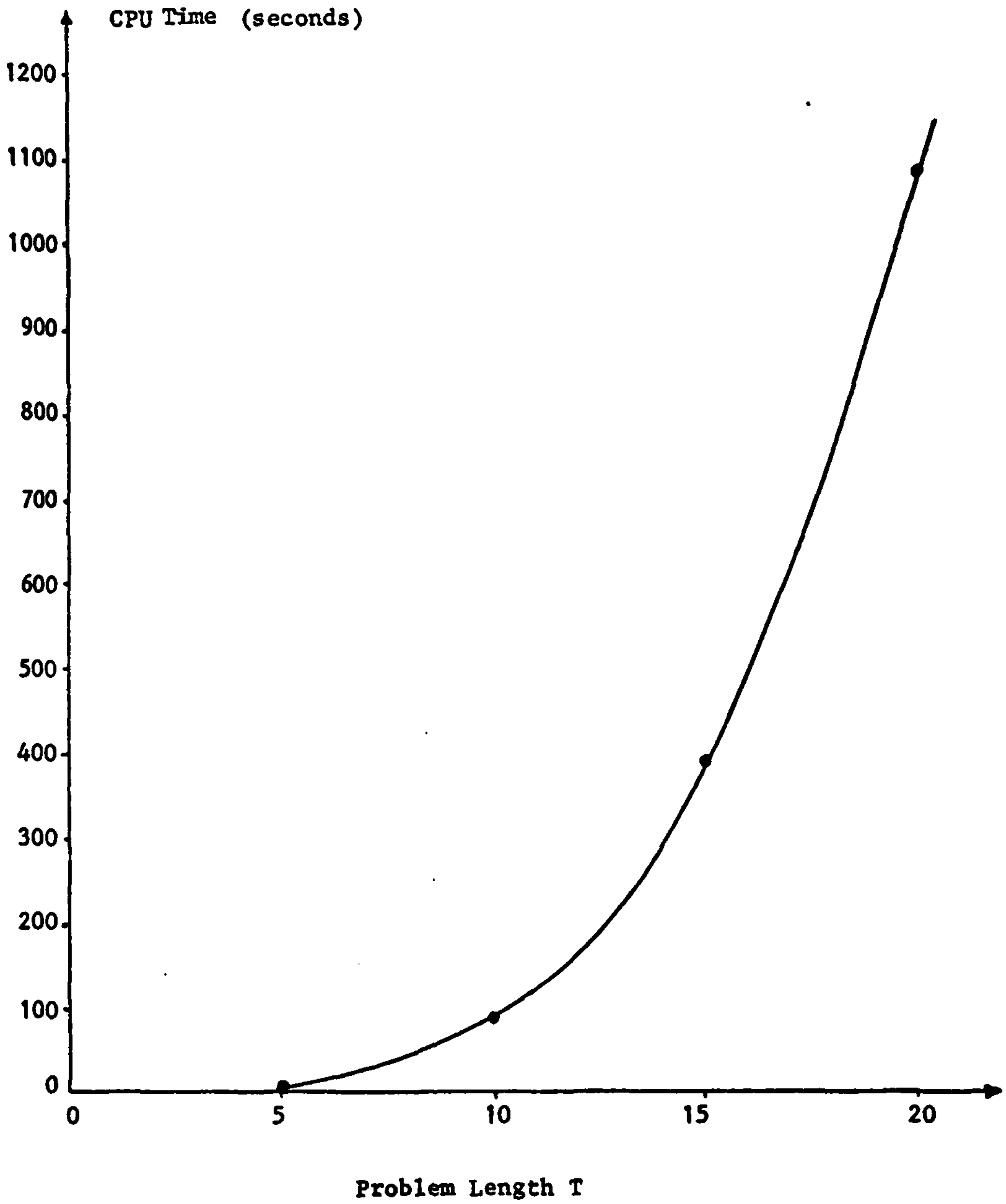


Figure 3: Mean Pivoting CPU Time versus T for 11 randomly generated manpower planning problems solved with STDLP

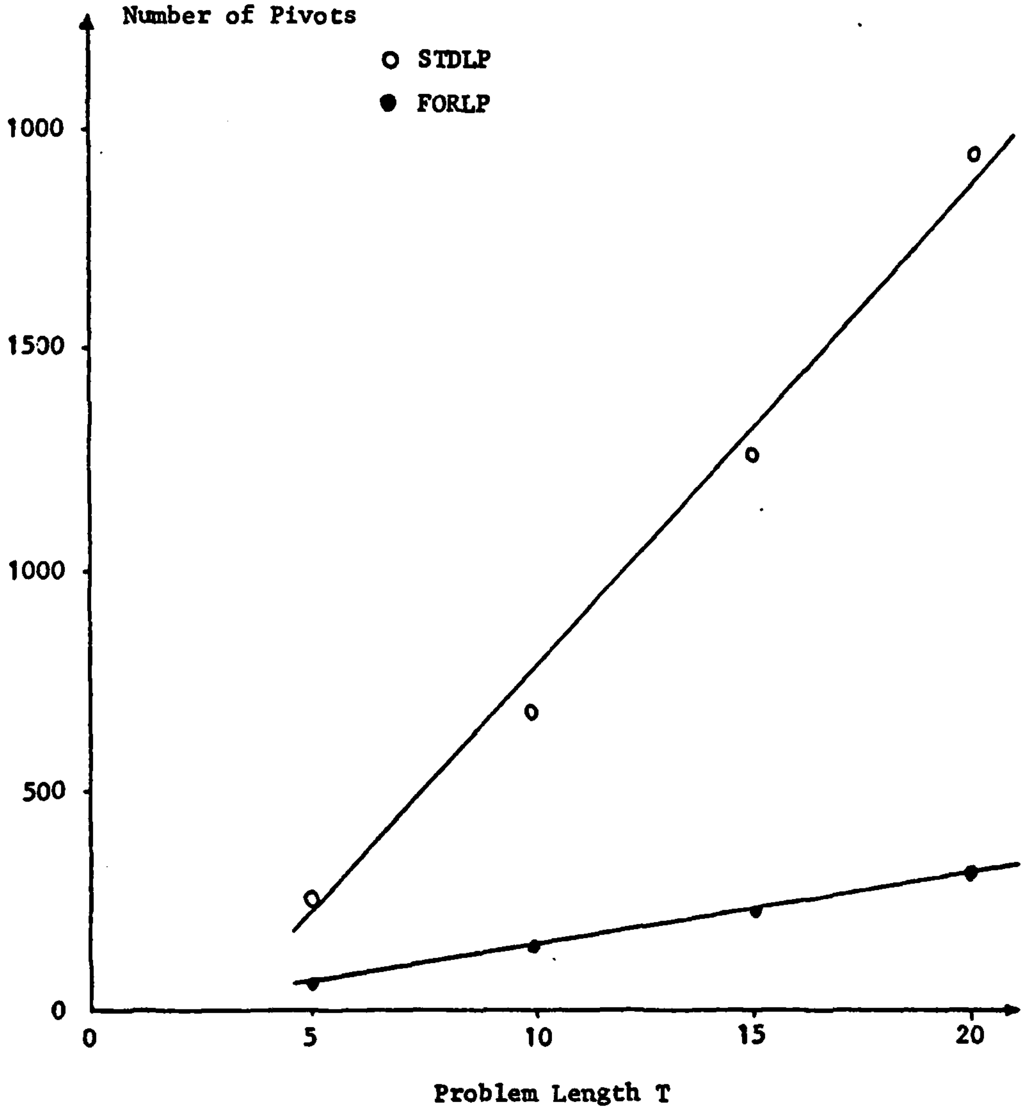


Figure 4: Mean Number of Pivots versus T for 11 randomly generated manpower planning problems solved with FORLP and STDLP

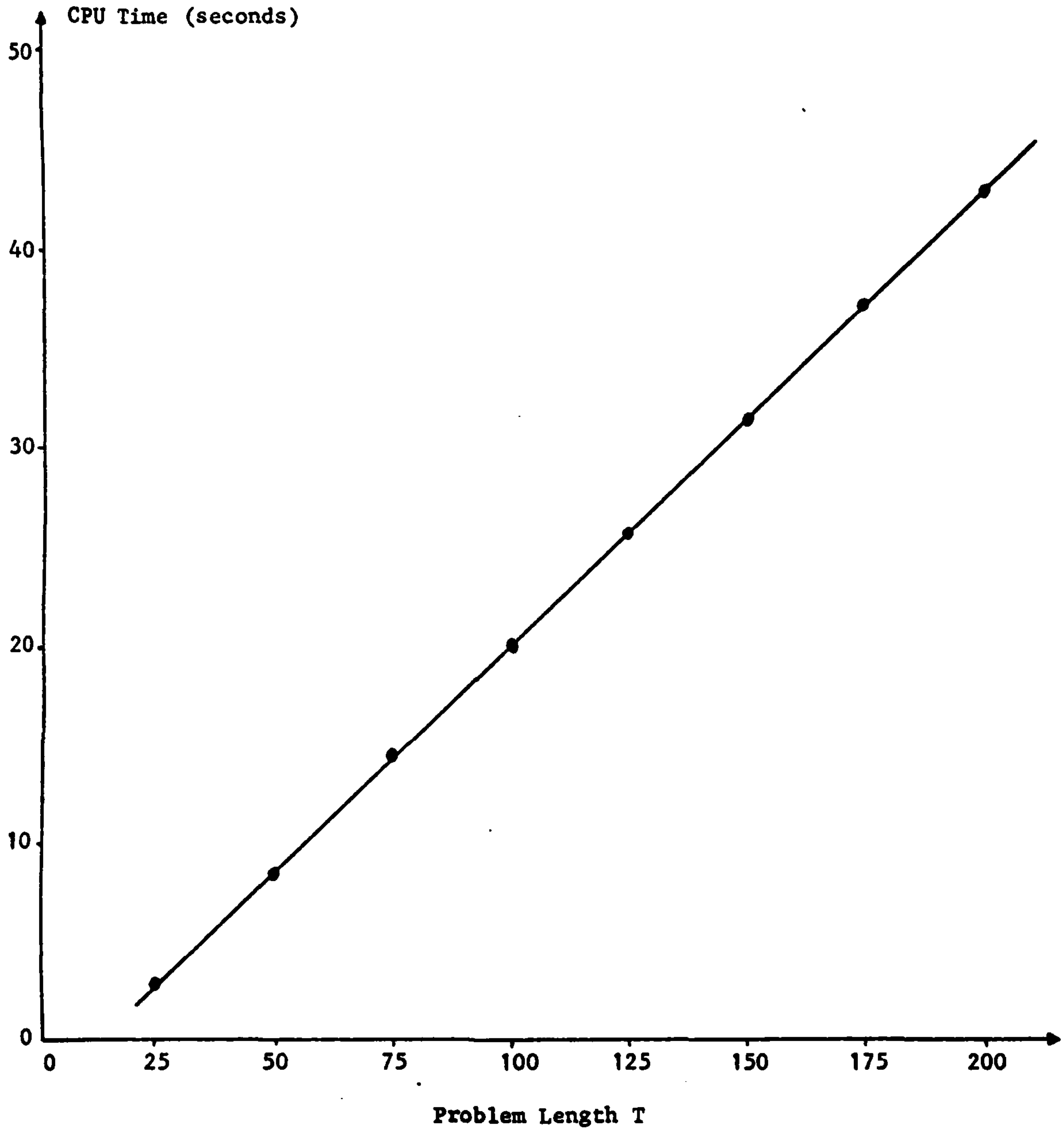


Figure 5: Mean Pivoting CPU Time versus T for 11 randomly generated manpower planning problems solved with FORLP

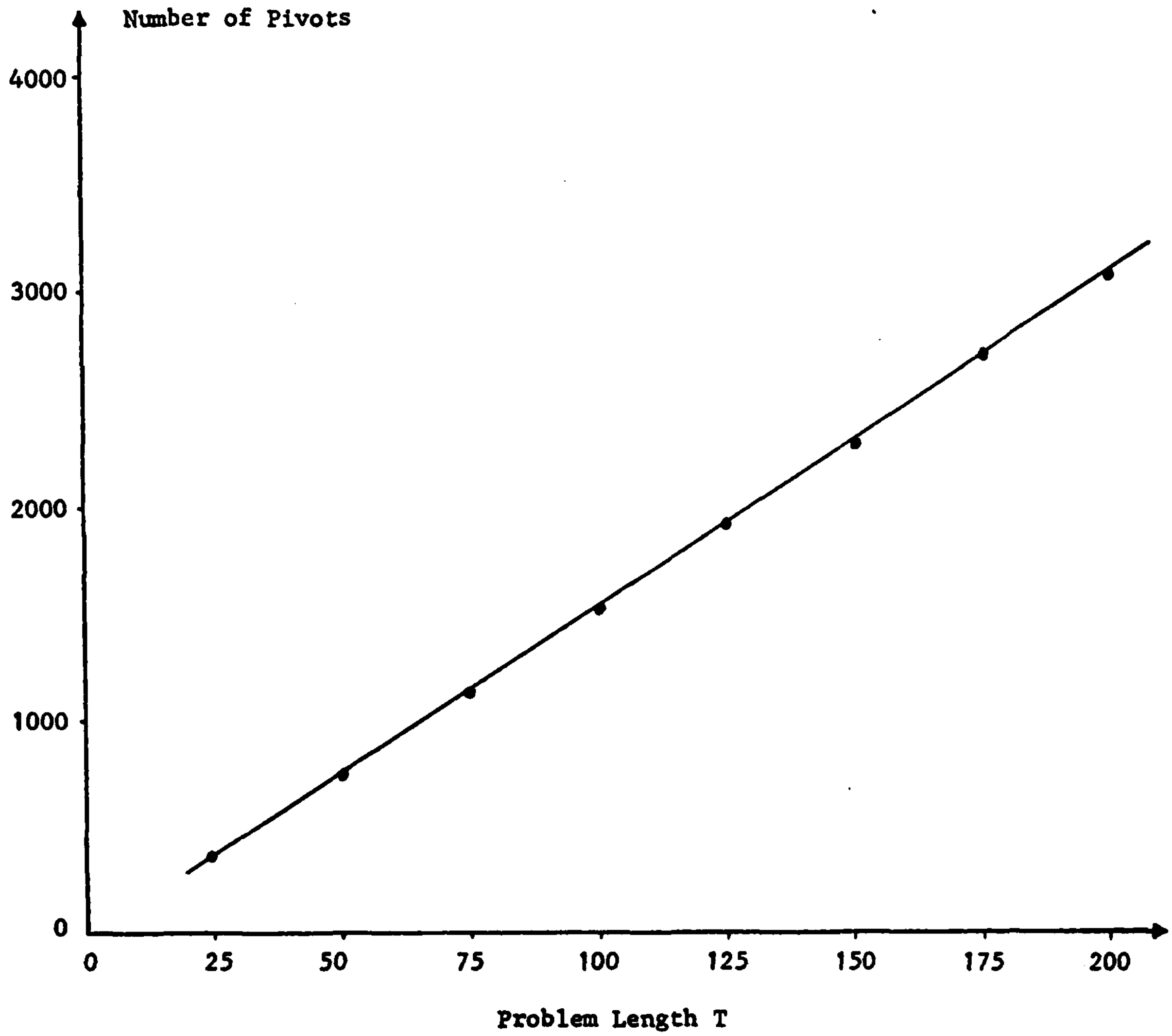


Figure 6: Mean Number of Pivots versus T for 11 randomly generated manpower planning problems solved with FORLP

7. Conclusions

The computational results here indicate that the Forward Simplex Method solution time is linear versus worse than cubic for a standard LP code. The Forward Simplex Method was able to solve problems more than ten times the size that the conventional code could accommodate. In fact, for problems that the conventional code could solve, it required 28 to 500 times as much pivoting CPU time to perform from 2 to 6 times the number of pivots that the Forward Simplex Method used.

The maintenance of the staircase structure and natural decomposition of large dynamic planning models are the features of the Forward Simplex Method which give its computational advantage. Thus, the extension of forward techniques to dynamic linear programs is a useful new method for solving staircase structure problems efficiently. It should be particularly useful for solving large multi-stage planning problems in a real-time conversational environment such as those described in [10, 11].

References

[1] Aronson, J.E., Forward Linear Programming, Ph.D. Thesis, Graduate School of Industrial Administration, Carnegie-Mellon University, Pittsburgh, Pennsylvania, 1980.

[2] Aronson, J.E., Morton, T.E., and Thompson, G.L., "A Forward Algorithm and Planning Horizon Procedure for the Production Smoothing Problem without Inventory," W.P. #20-78-79, Graduate School of Industrial Administration, Carnegie-Mellon University, Pittsburgh, Pennsylvania, November 1978.

[3] Aronson, J.E., Morton, T.E., and Thompson, G.L., "A Forward Simplex Method," W.P. #58-79-80, Graduate School of Industrial Administration, Carnegie-Mellon University, Pittsburgh, Pennsylvania, April 1980.

[4] Charnes, A., Cooper, W.W., Lewis, K.A., and Niehaus, R.J.,"Multi-level coherence models for equal employment opportunity planning,"TIMS Studies in the Management Sciences 8 (1978), 13-29.

[5] Charnes, A., Cooper, W.W., and Niehaus, R.J.,"Studies in Manpower Planning" (U.S. Navy Office of Civilian Manpower Management: Washington, D.C., 1972).

[6] Charnes, A., Cooper, W.W., and Niehaus, R.J.,"Dynamic multi-attribute models for mixed manpower systems,"Naval Research Logistics Quarterly, 22 (1975), 205-220.

[7] Crowder, H., Dembo, R.S., and Mulvey, J.M., "On Reporting Computational Experiments with Mathematical Software," ACM Transactions on Mathematical Software, 5, 2, June 1979, 193-203.

[8] Hellerman, E., and Rarick, D., "The Partitioned Preassigned Pivot Procedure (P^4)", in Sparse Matrices and Their Applications, Edited by Rose, D., and Willoughby, R., Plenum Press, New York, NY, 1972, 67-76.

[9] Hellerman, E., and Rarick, D., "Reinversion with the Preassigned Pivot Procedure," Mathematical Programming, 1, 1971, 195-216.

[10] Niehaus, R.J., Schultz, D., and Thompson, G.L., "Managerial Tests of Conversational Manpower Planning Models," TIMS Studies in the Management Sciences, 8, North-Holland Publishing Co., 1978, 153-171.

[11] Niehaus, R.J., Sholtz, D., and Thompson, G.L., "Guessing your way to your recruiting goals," The Journal of Navy Civilian Manpower Management (Spring 1973).

Unclassified

(9) Management science research rept.

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: M.S.R.R. 463

2. GOVT ACCESSION NO.: AD-A097 522

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle): (6) THE SOLUTION OF MANPOWER PLANNING PROBLEMS BY THE FORWARD SIMPLEX METHOD.

5. TYPE OF REPORT & PERIOD COVERED: Technical Report

6. PERFORMING ORG. REPORT NUMBER: M.S.R.R. 463

7. AUTHOR(s): (10) Jay E. Aronson, G. L. Thompson, Gerald

(12) 20

8. CONTRACT OR GRANT NUMBER(s): (15) N00014-75-C-0621

9. PERFORMING ORGANIZATION NAME AND ADDRESS:
Graduate School of Industrial Administration
Carnegie-Mellon University
Pittsburgh, Pennsylvania 15213

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: WR-6-0147

11. CONTROLLING OFFICE NAME AND ADDRESS:
Personnel and Training Research Programs
Office of Naval Research Code (458)
Arlington, Virginia 22217

12. REPORT DATE: (11) Aug [redacted]80

13. NUMBER OF PAGES: 16

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office): (14) MSRR-463, WP-38-80-81

15. SECURITY CLASS. (of this report)

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Forward simplex method, Staircase structured linear programs, Large scale linear programs.

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The use of the forward simplex algorithm of Aronson, Morton, and Thompson to solve the multi-stage personnel planning linear programming models of Charnes, Cooper and Niehaus is described. Computational results on randomly generated problems having up to 200 periods indicate that the forward simplex method requires CPU time and number of pivots which are linear in the number of periods. The standard simplex method requirements vary with at least the cube of the number of periods. For this reason the forward simplex method should be especially useful for solving real-time, conversational versions of personnel (and other) planning models.

DD FORM 1473, 1 JAN 73 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601

Unclassified
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

403426